AD-A114 647    HARVARD UNIV  CAMBRIDGE MA AIKEN COMPUTATION LAB        F/G 12/1
               THE COMPLEXITY OF TWO PLAYER GAMES OF INCOMPLETE INFORMATION.(U)
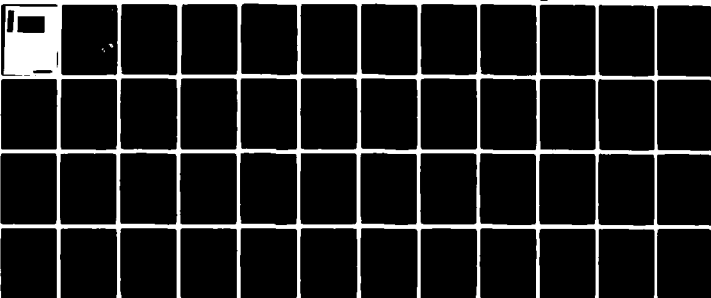               OCT 81   J H REIF                                   N00014-80-C-0647
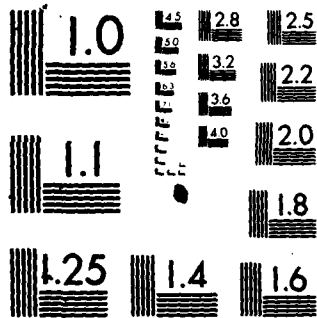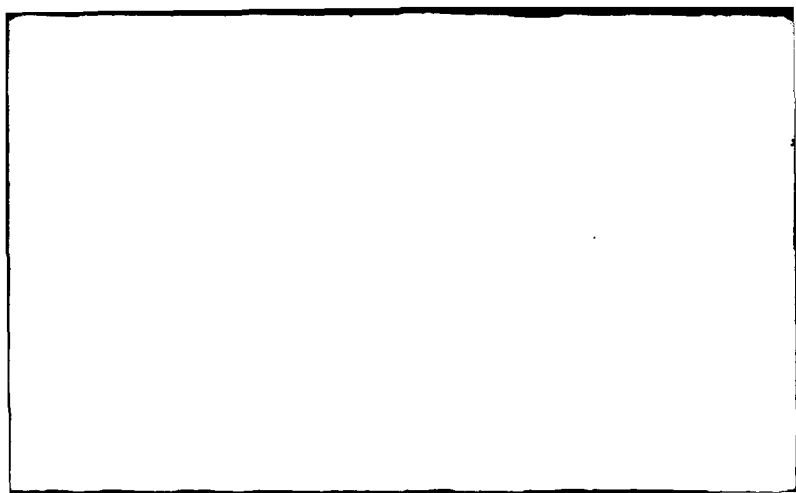UNCLASSIFIED   TR-35-81                                            NL

END
DATE
FILMED
6—82
DTIC

1.0

1.1

1.25  1.4  1.6

4.5
50
5.6

2.8  2.5
3.2  2.2
3.6
4.0  2.0

1.8

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

| REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|

| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
|---|---|---|
| | AD-A114647 | |

| 4. TITLE (and Subtitle) | 5. TYPE OF REPORT & PERIOD COVERED |
|---|---|
| The Complexity of Two Player Games of Incomplete Information | Technical Report |
| | 6. PERFORMING ORG. REPORT NUMBER TR-35-81 |

| 7. AUTHOR(s) | 8. CONTRACT OR GRANT NUMBER(s) |
|---|---|
| John H. Reif | N00014-80-C-0647 |

| 9. PERFORMING ORGANIZATION NAME AND ADDRESS | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
|---|---|
| Harvard University Cambridge, MA 02138 | |

| 11. CONTROLLING OFFICE NAME AND ADDRESS | 12. REPORT DATE |
|---|---|
| Office of Naval Research 800 North Quincy Street Arlington, VA 22217 | October, 1981 |
| | 13. NUMBER OF PAGES 53 |

| 14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office) | 15. SECURITY CLASS. (of this report) |
|---|---|
| same as above | |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

**16. DISTRIBUTION STATEMENT (of this Report)**

unlimited

**17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, If different from Report)**

unlimited

**18. SUPPLEMENTARY NOTES**

**19. KEY WORDS (Continue on reverse side if necessary and identify by block number)**

two-player games, incomplete information, decision algorithms, universal games, private alternation, blindfold games, blind alternation, computational complexity.

**20. ABSTRACT (Continue on reverse side if necessary and identify by block number)**

see reverse side

82 05 19 027

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) The Complexity of Two Player Games of Incomplete Information | | 5. TYPE OF REPORT & PERIOD COVERED Technical Report |
| | | 6. PERFORMING ORG. REPORT NUMBER TR-35-81 |
| 7. AUTHOR(s) John H. Reif | | 8. CONTRACT OR GRANT NUMBER(s) N00014-80-C-0647 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Harvard University Cambridge, MA 02138 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 North Quincy Street Arlington, VA 22217 | | 12. REPORT DATE October, 1981 |
| | | 13. NUMBER OF PAGES 53 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) same as above | | 15. SECURITY CLASS. (of this report) |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

two-player games, incomplete information, decision algorithms, univ
games, private alternation, blindfold games, blind alternation,
computational complexity.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see reverse side

THE COMPLEXITY OF TWO PLAYER GAMES
OF INCOMPLETE INFORMATION

John H. Reif

TR-35-81

October, 1981

The Complexity of Two Player Games

of Incomplete Information*

by

John H. Reif[+]

October, 1981

# SUMMARY

We consider two-player games of *incomplete information* in which certain portions of positions are private to each player and cannot be viewed by the opponent. We provide asymptotically optimal decision algorithms for space bounded games. We present various games of incomplete information which are shown to be *universal* in the sense that they are the hardest of all reasonable games of incomplete information. The problem of determining the outcome of these universal games from a given initial position is shown to be complete in doubly-exponential time. We also define "private alternating Turing machines" which are a new type of alternating Turing machines with certain private types. The space complexity $S(n)$ of these machines is characterized in terms of the complexity of deterministic Turing machines, with time bounds doubly-exponential in $S(n)$.

We also consider *blindfold games*, which are restricted games in which the second player is not allowed to modify the common position. We provide asymptotically optimal decision algorithms for space bounded blindfold games. We also show various blindfold games to have exponential space complete outcome problems and to be universal for reasonable blindfold games. We define "blind alternating Turing machines" which are private alternating Turing machines with restrictions similar to those in blindfold games. A single exponential jump characterizes the relation between the space complexity of these machines and the space complexity of deterministic Turing machines.

## 1. INTRODUCTION

A (two-player) game  G  consists essentially of disjoint sets of positions for two players named 0 and 1, plus relations specifying legal next-moves for the players.  We assume positions are strings over a finite alphabet.  A *position*  P  contains portions which are *private* to each player (invisible to their opponent) and the remaining portions of P  are *common* and may be publicly viewed by both players.  The set of legal next-moves for a given player must be independent of the opponent's private portions  of positions.

The game  G  is of *perfect information* if no position contains a private portion.  On the other hand, a game is *blindfold* if player 2 never modifies the common portion of a position.

For example, consider the game ‎PEEK‎ of Figure 1a.  (PEEK was first described in [Chandra and Stockmeyer, 1978].)  A *position of* PEEK consists of a box with two open ends and containing various plates stacked horizontally within.  The plates are perforated by holes of uniform size in various places.  The top and bottom of the box are also perforated with holes.  Each plate contains a knob on one of the open ends of the box, and the plate may slide to either of two locations "in" or "out".  Once "out", a plate can only be pushed "in", and *vice versa*.  The players stand at the two open ends of the box.  A *move* by a player  $i \in \{0,1\}$ consists of grasping a knob from his side and pushing the corresponding plate either "in" or "out".  The player may also pass.  If just after the move there is a hole in each plate lined up vertically (so the player can "peek" through from the top to the bottom), then *player i wins*.  The game PEEK is of

perfect information: each player knows the pattern of holes on the plates and can view the location of all the plates.

To introduce *private portions of positions*, we place partial barriers on both ends of the box, as in Figure 1b. These barriers hide the location of some, but perhaps not all, of the opponent's plates. Nevertheless, both players are still aware of the pattern of holes in the plates and can attempt to "peek" through the box from the top. Let PRIVATE-PEEK be the resulting game of incomplete information. By requiring that the barriers on the side of player 1 obscure the locations of *all* the opponent's plates, we have the blindfold game BLIND-PEEK (see Figure 1c).

The *outcome problem for a game* G is the problem of determining the existence of a winning strategy for player 1, given an initial position. If no *a priori*-bound is placed on the size of positions of games, the outcome problem is undecidable (see the computation games of Section 3). We consider a game to be *reasonable* if its space bound for positions is $O(n)$.

Given a class of games $\mathscr{C}$, a game G is *universal* to $\mathscr{C}$ if (1) $G \in \mathscr{C}$ and (2) the outcome problem for each $G' \in \mathscr{C}$ is log-space reducible (see [Stockmeyer and Meyer, 1973]; a log-space reduction is always polynomial time) to the outcome problem for G. The game PEEK was shown universal to reasonable games of perfect information in [Stockmeyer and Chandra, 1979]. We show BLIND-PEEK is universal for all reasonable blindfold games, and that PRIVATE-PEEK is universal for all reasonable games. While the outcome problem for PEEK is (log-space) complete in exponential time, the outcome problem for BLIND-PEEK is complete in exponential space, and the outcome problem for PRIVATE-PEEK is complete in double exponential time.
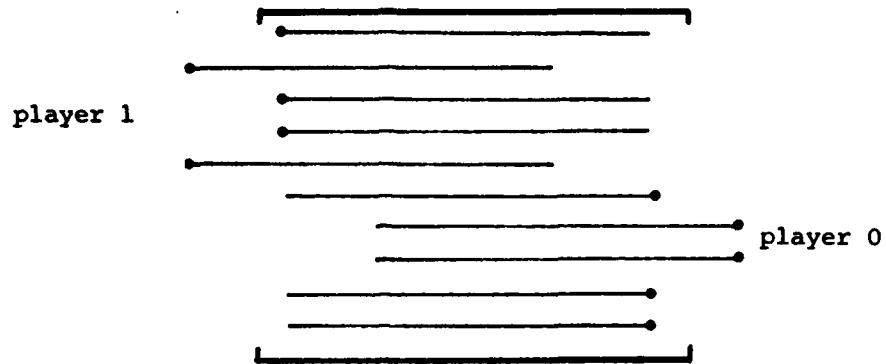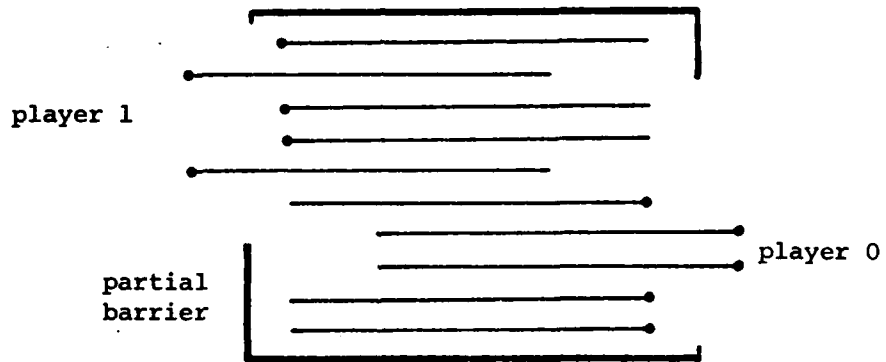
Figure 1a:  A position of PEEK



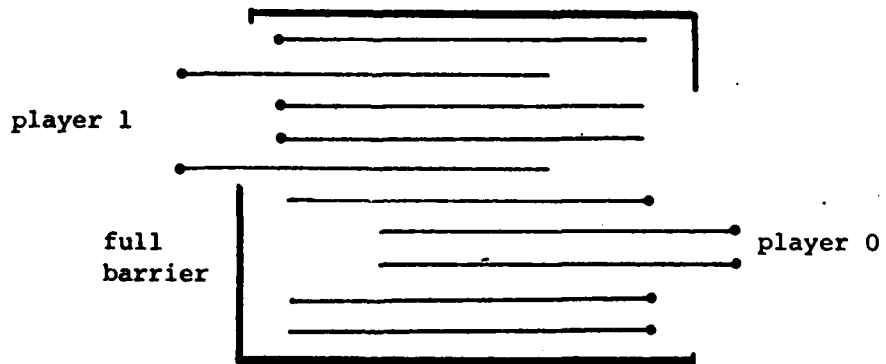Figure 1b:  A position of PRIVATE-PEEK



Figure 1c:  A position of BLIND-PEEK

Games (with easy-to-compute next-move relations) can be considered to be computing machines. Game G *accepts input* ω, depending on the outcome of the game from an initial position containing ω. Games of perfect information related in this way to the *alternating machines* (A-TMs) of [Chandra, Kozen and Stockmeyer, 1978] in which existential states (identified with player 1) alternate with universal states (player 2) during a computation. Also, *nondeterministic Turing Machines* (N-TMs) are related to games of perfect information with the second player absent, and *deterministic Turing Machines* (D-TMs) are related to a similar class of games with a deterministic next move relation.

In this paper we introduce two new types: *private* and *blind alternating machines*. We add to an A-TM certain work tapes private to universal states (player 2); the machine cannot read the private tapes while in existential states. The result is a private alternating machine (PA-TM). (See Figure 2). For a blind alternating machine (BA-TM) we restrict a PA-TM so that the universal states can write only on their private tapes, and on no other tapes. *Acceptance* of input strings by these machines is defined by the outcome in corresponding computation games.

Let $\mathscr{F}$ be a set of functions on variable n. For each $\alpha \in \{D,N,A,PA,BA\}$, let $\alpha SPACE(\mathscr{F})$ be the class of languages accepted by $\alpha$-TMs within some space bound in $\mathscr{F}$, and let $\alpha TIME(\mathscr{F})$ be the class of languages accepted by $\alpha$-TMs within some time bound in $\mathscr{F}$. Let $EXP(\mathscr{F})$ be the set of functions

$$\{c^{F(n)} \mid c > 0 \quad \text{and} \quad F(n) \in \mathscr{F}\}$$

We drop the set brackets in the above notation if $\mathscr{F}$ is a singleton set and let $EXP(f(n))$ denote $EXP(\{f(n)\})$. For example, the polynomial

read only input tape

finite states

existential states | universal states
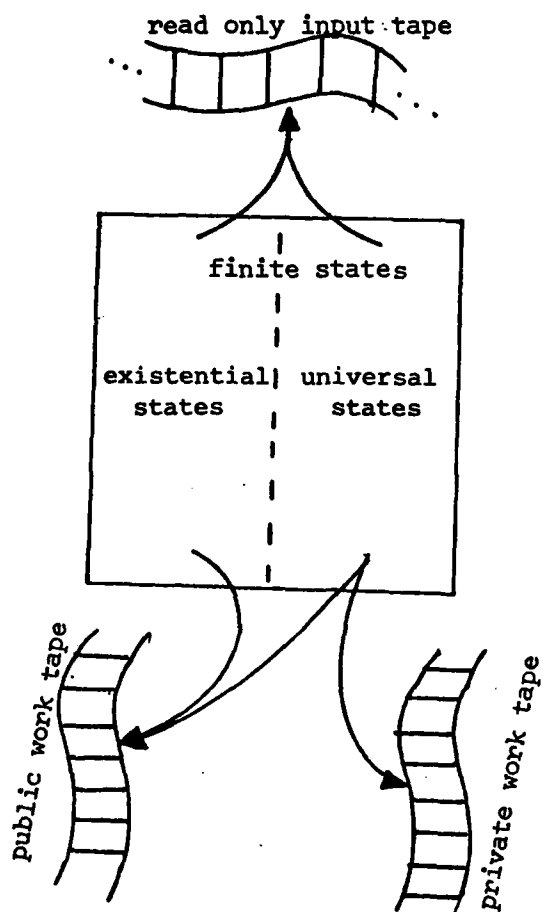
public work tape

private work tape

figure 2:  An alternating Turing machine with a tape private to the universal states

functions $POLY(n) = \{n^c | c \geqslant 1\}$ can be defined in this notation as $POLY(n) = EXP(\log n)$.

[Chandra, Kozen and Stockmeyer, 1978] relate the space and time complexity of A-TMs and D-TMs as follows:

For each function $S(n) \geqslant \log n$ and function set $\mathscr{F} = EXP(S(n))$,

$$ASPACE(S(n)) = DTIME(\mathscr{F})$$

$$ATIME(EXP(S(n))) = DSPACE(\mathscr{F}).$$

We characterize the space complexity PA-TMs and BA-TMs in terms of the time and space complexity of A-TMs and D-TMs as follows:

For each function $S(n)$ and function set $\mathscr{F}$ as above,

$$BASPACE(S(n)) = ATIME(\mathscr{F})$$
$$= DSPACE(\mathscr{F})$$
$$PASPACE(S(n)) = ASPACE(\mathscr{F})$$
$$= DTIME(EXP(\mathscr{F})).$$

Also, the time complexity of PA-TMs, BA-TMs and A-TMs are all roughly the same. For functions of $\mathscr{F} = EXP(S(n))$ as above

$$PATIME(\mathscr{F}) = BATIME(\mathscr{F})$$
$$= ATIME(\mathscr{F})$$
$$= DSPACE(\mathscr{F}).$$

This paper is organized as follows: the next section defines games of incomplete information; Section 3 introduces our PA-TMs and BA-TMs; Section 4 presents decision algorithms for space bounded games, and also games with both alternation and space bounds; Section 5 gives lower bounds on the complexity of space bounded games; Section 6 considers the complexity of time bounded PA-TMs and BA-TMs; Section 7 describes certain propositional formula games which are universal for reasonable games; and Section 8 concludes the paper with mention to extensions of this work to multi-player games and multiprocessing, in collaboration with Gary Peterson.

TIME                                    SPACE
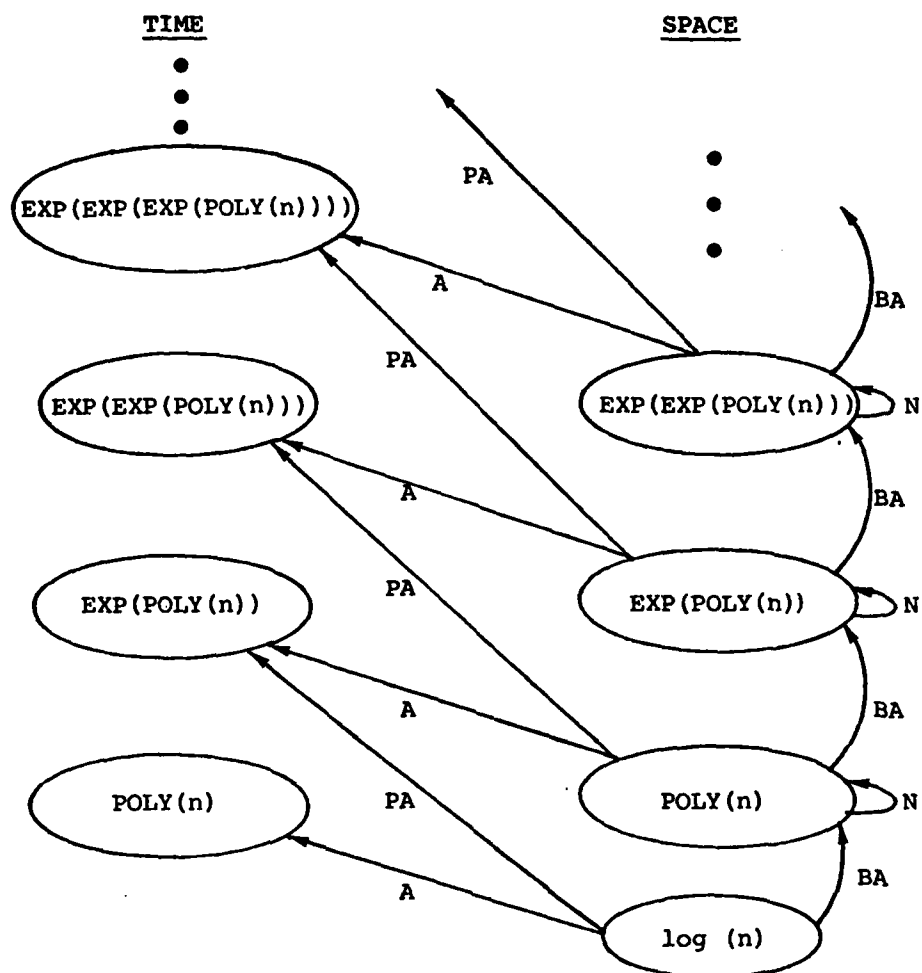


Figure 3:   Complexity Jumps for α-TMs from α-SPACE to deterministic time
            and space.

α = A   for "alternation
  = BA  for "blind alternation"
  = PA  for "private alternation"
  = N   for "nondeterministic"

## 2. TWO PLAYER GAMES OF INCOMPLETE INFORMATION

### 2.1 Game Definitions

A (two player) *game* is a tuple $G = (POS, \vdash)$ where

(i) POS is the set of *positions*, with $POS = \{0,1\} \times PP_0 \times PP_1 \times CP$ and $PP_0$, $PP_1$, CP are sets of strings over a finite alphabet.

(ii) $\vdash \subseteq POS \times POS$ is the *next move relation* and $\vdash$ satisfies axioms A1, A2 given below.

The *player* are named 0, 1. If $p = (i, pp_0, pp_1, cp)$ is a position in POS, then p is composed of a number $i \in \{0,1\}$ indicating which player's turn is next, a portion $pp_0$ which is *private to player* 0, a portion $pp_1$ which is *private to player* 1, and a *common portion* cp.

For $i \in \{0,1\}$, let $POS_i$ be the set of positions with 1st component i; thus $POS_i$ are the positions for which it is player i's next move.

Informally, a player *wins* by making the last move. Thus the object of the game is to force the opponent into a position from which there is no next move. Formally, let the set of *winning positions* be $W = \{p \in POS \mid p \vdash p'$ for no $p'\}$. If $p \in W \wedge POS_0$ then p is a *winning position* for player 1.

Given a position $p = (i, pp_0, pp_1, cp)$ let $vis_1(p) = (i, pp_1, cp)$ be the *portion visible to player* 1 and let $priv_1(p) = pp_1$ be the *portion of* p *private to player* 1 ($vis_0(p)$ and $priv_0(p)$ are defined similarly, with 0 in place of 1).

The idea of imperfect information is captured in the following two axioms. Axiom 1 asserts that a player can't modify the portion of the position private to his opponent. Axiom 2 asserts that a player's possible moves next are independent of the portion of the position private to his opponent.

**A1:**    If $p \in POS_1$ and $p \vdash p'$ then $priv_0(p) = priv_0(p')$.

**A2:**    If $p, q \in POS_1 - W$ and $vis_1(p) = vis_1(q)$ then

$$\{vis_1(p') \mid p \vdash p'\} = \{vis_1(q') \mid q \vdash q'\}.$$

We also assume both A1 and A2 hold with 0 exchanged with 1.


## 2.2  Plays and Strategies for Games

For any finite string $\pi$ of positions, let $last(\pi)$ be the last position of $\pi$. Fix an *initial position* $p_I \in POS$. A *play* is a (possibly infinite) string $\pi = p_0 p_1 \ldots$ of positions such that $p_0 = p_I$ is the initial position, $p_0 \vdash p_1$, $p_1 \vdash p_2, \ldots$ and $last(\pi) \in W$ whenever $\pi$ is finite. A play $\pi$ is said to be a *win for player* 1 if $\pi$ is finite and $last(\pi) \in POS_0 \wedge W$.

A *play prefix* $\pi$ is a finite nonnull initial substring of a play. Intuitively, a play prefix represents a sequence of legal moves starting from initial position. Note that the players need not alternate. After any play prefix $\pi$, the sequence $vis_1(\pi)$ represents the extent of player 1's knowledge about the game play to date. We define $vis_1(\pi)$ inductively. Let $p = last(\pi)$. If $\pi$ is of length 1 then $vis_1(\pi) = vis_1(p)$. Suppose we are given $p' \in POS$ such that $p \vdash p'$. If both $p \in POS_0$ and $vis_1(p) = vis_1(p')$ then $vis_1(\pi p') = vis_1(\pi)$, (intuitively, this first case insures that player 1 cannot detect moves of player 0 which do not modify that portion of $p$ visible to player 1), and otherwise $vis_1(\pi) = vis_1(\pi), vis_1(p)$.

The *game tree* $T$ is the set of play prefixes. The *root* of $T$ is the initial position $p_I$. Each play prefix $\pi$ is considered a *node* of $T$.

The *children* of $\pi$ are those play prefixes $\pi'$ of length one more than $\pi$ and such that $\pi$ is a prefix of $\pi'$. Let $T_1$ be the set of play prefixes $\pi$ such that $last(\pi) \in POS_1 - W$; thus it is player 1's turn to move at $last(\pi)$.

A *strategy* for player 1 is a function $\sigma: T_1 \to T$ such that

(1)  for any $\pi \in T_1$, $\sigma(\pi)$ is a child of $\pi$, and

(2)  if $\pi, \pi' \in T_1$ and if $vis_1(\pi) = vis_1(\pi')$ then $vis_1(\sigma(\pi)) = vis_1(\sigma(\pi'))$.

Thus $\sigma$ is a rule for player 1 to select his next move. Condition (2) says that this selection must be made only on the basis of the knowledge player 1 has about the progress of the game to date. A play $\pi$ is a *play by strategy* $\sigma$ if whenever $\pi'$ is a prefix of $\pi$ and $\pi'$ is in the domain of $\sigma$, then $\sigma(\pi')$ is a prefix of $\pi$. $\sigma$ is called a *winning strategy* for player 1 iff every play by strategy $\sigma$ is a win for player 1.

The *outcome problem* for game $G$ is:

given an initial position $p_I \in POS$, is there a winning strategy for player 1?

Note that although games such as Checkers and Go have standard initial positions, their rules may be readily generalized to $n \times n$ bounds. Initial positions of games are not always fixed in this paper, since we shall wish to consider the outcome problem for games, given *arbitrary* initial positions. This allows us a meaningful notion of the complexity of the outcome of these games. The *complexity of* various generalized games of perfect information is considered in [Schaefer, 1978], [Evan and Tarjan, 1976], [Frankel, Garey and Johnson, 1978], [Lichtenstein and Sipser, 1980 and 1981], [Stockmeyer and Chandra, 1979]. The complexity of a blindfold game was first considered in [Jones, 1978].

To model a game like two player poker, in which players do not have perfect information even at the start, we may simply add an initial move which allows player 0 to choose both player's cards; this is justified by Proposition 2.1, given below.

It should be clear that the outcome of a game is not effected if player 0 is allowed to "cheat", by viewing the private portions of player 1's positions. For each position $p \in POS$, let $p^c$ be the position derived from $p$ by making common to *both* players that portion of $p$ originally private to player 1. Let $G^c$ be the game so derived from game $G$. It follows immediately from our definition of strategies that

PROPOSITION 2.1. Player 1 has a winning strategy in $G$ from initial portion $p_I$ iff player 1 has a winning strategy in $G^c$ from initial position $p_I^c$.

Nevertheless, the outcome of *probabilistic strategies*, as defined in [Reif, 77] are highly dependent on the existence of private positions of both players.


## 2.3 Special Types of Games

A game is *perfect information* if the private portions of any position is the fixed value *null*, so that the only information of interest in a position is the common portion. Thus $POS \approx \{0,1\} \times CP$. For example, Chess, Checkers, and Go are all games of perfect information.

A strategy $\sigma$ is *Markov* if $\sigma(\pi) = \sigma(\pi')$ for all play prefixes $\pi$, $\pi'$ such that $last(\pi) = last(\pi')$. Markov strategies are independent of previous play except for the current position. Thus it suffices to

to consider a Markov strategy to be a mapping from the current position to the next position.

PROPOSITION 2.2.  In any game of perfect information, if player 1 has a winning strategy $\sigma$, then player 1 has a winning Markov strategy.

To prove this proposition we observe that for any play prefixes $\pi$, $\pi'$ such that $last(\pi) = last(\pi')$, we can define strategy $\sigma'$ identical to $\sigma$ except that $\sigma'(\pi) = \sigma(\pi)$.  Then $\sigma'$ is winning for player 1 if $\sigma$ was.  Repeating this process yields a winning Markov strategy.  On the other hand, strategies for games of incomplete information must generally depend on previous play to determine the possible private positions of the opponent.

A game is *blindfold* if $vis_1(p) = vis_1(p')$ whenever $p \in POS_0$ and $p \vdash p'$; thus there is no interchange of information between players in blindfold games.  Some examples of blindfold games are given in Section 7. Also see [Jones, 1978].  The German game of "blind chess" is not truly a blindfold game since there is a gradual transfer of positional knowledge.

A game is *solitaire* if $p \vdash p'$ for any $p \in POS$ then $p' \in POS_1 \cup W$. Thus player 0 does not enter into the play after the first move.  An initial move of player 0, preceding any move of player 1, allows player 0 to develop its private portion of the position.  For example, Battleship, Mastermind, and of course the card game of Solitaire are all solitaire games.

A game is *nondeterministic* if $POS_0 \wedge (POS-W)$ is empty.  Note that a nondeterministic game can always be made a game of perfect information without modifying its outcome, by simply letting the private portions of positions be in the common portion of positions.

Finally, a game is *deterministic* if the next move relation ⊢ is; i.e., for each portion $p \in POS$, there is at most one position $p' \in POS$ such that $p \vdash p'$.

## 2.4 Complexity Bounds on Games

Let $G = (POS, \vdash)$ be a game. Let us assume for any position $p \in POS$, the positions $\{P' | p \vdash p'\}$ are ordered $\vdash_1(p), \ldots, \vdash_d(p)$ so that $\vdash_i(p)$ is the i'th position derived by a next move from p. We assume a *next move transducer for* ⊢; this is a Turing Machine NM with two distinguished read only input tapes, a write only output tape, and possibly some work tapes. In general, we assume NM is in configuration $I(p)$, where either $p = p_I$ is the initial position, or p is the last position just output by NM. In either case, we assume the first input tape contains the initial position $p_I$, and the second input tape contains an integer $i \geqslant 1$. Then NM writes position $\vdash_i(p)$ on the output tape (NM outputs the empty string T if the i-th position derived from p does not exist). Thus NM may be used to generate the positions of any play prefix. (NOTE the input/output tapes of NM are not charged for space.)

Let a move $p \vdash p'$ be an *alternation* if $p' \notin W$ and either ($p \in POS_0$ and $p' \in POS_1$) or ($p \in POS_1$ and $p' \in POS_0$).

Game G has *time bound* $T(n)$ (*alternation bound* $A(n)$, *space bound* $S(n)$, respectively) if on each position $p_I \in POS$ of length n from which player 1 has a winning strategy $\sigma$, there is some such $\sigma$ where $\pi$ has $\leqslant T(n)$ moves ($\pi$ has $\leqslant A(n)$ alternations, the next move transducer requires $\leqslant S(n)$ work tape calls for the moves of $\pi$, respectively), for each play $\pi$ induced by $\sigma$ with initial position $p_I$.

It is interesting to note that any game with a fixed initial position and finite time or space bound, can be represented as a physical object with a finite game board and a finite set of tokens for marking positions.

Let  G  be a *reasonable game* if it has space bound  $O(n)$.

## 3. PRIVATE AND BLIND ALTERNATING MACHINES

The *alternating machine* proposed by [Chandra, Kozen and Stockmeyer, 1979] has a natural correspondence to games of perfect information. The states of alternating automata are named either *universal* or *existential*. The sequencing between existential and universal states corresponds to the alternation of moves by players in the play of game.

We introduce here a new type of alternating machine with private tapes which have a natural correspondence to games of incomplete information. In fact, we will define the languages accepted by these machines by the existence of winning strategies for the corresponding computation games.

Let a *private alternating machine* (PA-TM) be a tuple

$$M = (Q, Q_u, q_I, \Sigma, \Gamma, \#, b, t, t_p, \delta)$$

where

$Q$ is a finite set of *states*

$Q_u \subseteq Q$ are the *universal states*

$(Q - Q_u$ are the *existential states*)

$q_I \in Q$ is the *initial state*

$\Sigma, \Gamma$ are the finite sets of *input* and *tape* symbols with $\Sigma \subseteq \Gamma$

$\#, b \in \Gamma - \Sigma$ are the distinguished *endmarker* and *blank* symbols,

$t$ is the number of tapes and $t_p$ is the number of private tapes

$\delta \subseteq (Q \times \Gamma^t) \times (Q \times \Gamma^t \times \{\underline{\text{left}}, \underline{\text{right}}, \underline{\text{static}}\}^t)$ is the *transition relation*, with restrictions given below.

There is a read-only *input tape*. Initially the input tape contains ¢w¢, with the input tape head scanning the first symbol of w, where w ∈ Σ* is an input string. (We assume there are no transitions past the endmarkers ¢.) There are also t - 1 *work tapes*, initially containing two-way infinite strings of the blank symbol b. The tapes $1, \ldots, t_p$ are *private work tapes*; they can only be written on from a universal state, and the transitions from each existential state are independent of the contents of the private work tapes (these restrictions to δ are made precise below). The other $t_0 - t_p - 1$ tapes are *common work tapes* and might be written on from any state of Q. The *contents* of a tape are given as (L, R) where L is the nonblank suffix of the portion of the tape to the left of the scan head, and R is the nonblank prefix of the portion of the tape just under and to the right of the scan head.

We now define the *computation game* $c^M$ = (POS,⊢) where POS are the positions (to be defined) of M and the next moves ⊢ ⊆ POS × POS are as defined by the transition relation δ of M. The player 1 is called the *existential player* and the player 0 is called the *universal player*.

Let a *position* of M be a tuple $p = (i, pp_0, pp_1, cp)$ where

(i)   $i \in \{0,1\}$ indicates that the current state is either existential (i = 1) or universal (i = 0)

(ii)  the *portion* $pp_0$ *private to the universal player* contains the contents of the private tapes,

(iii) the portion $pp_1$ private to the existential player 1 is *null*.

(iv)  the *common portion* cp is a pair whose first part is the current state, and whose second part is the contents of the common work tapes.

Thus the portion $vis_1(p)$ *visible to the existential player* is all of p
but the contents of the private tapes, and the portion $vis_0(p)$ *visible
to the universal player* is all of p. (This is justified by
Proposition 2.1.) We require $G^M$ to satisfy axioms A1, A2; this gives
us our required restricts on the transition function $\delta$ of M.
(NOTE. We may also decompose each of the states into a *common component*
and a *private component*, with restrictions to the transition relation just
as given here for the private tapes. This additional complication given
in our original [Reif, 1979] definition of PA-TMs is not·required as long
as we have at least one cell of one private tape; which may be used to
store the state of the universal player.)

For any input string $w \in \Sigma^*$, let the *initial position* $p_0(w)$ have
initial state $q_I$ and tape contents initialized as described above. We
introduce some (redundant) terminology to aid the reader's intuition.
The *accepting states* are those universal states with no successors. The
*rejecting states* are those existential states with no successors. Each
play of $G^M$ is called a *computation sequence* and the game tree T
called a *computation tree*. The input string $w \in \Sigma^*$ is *accepted by* M
if the existential player has a winning strategy. The computation
sequences induced by a winning strategy form an *accepting subtree* of T.
Let the *language of* M be $L(M) = \{w \in \Sigma^* | w$ is accepted by M$\}$.

The PA-TM is a natural·generalization of various *types* of machines
previously described in the literature. If M has no private tapes, it
is an *alternating machine* (A-TM) as described by [Chandra, Kozen and
Stockmeyer, 1979]. These have computation games where are of perfect
information. If M is further restricted to allow only those universal

states which are accepting (i.e., have no successors), then it is a
*nondeterministic Turing machine* (N-TM) as is now common in the literature.
If the transition relation of M is still *further restricted to be*
deterministic, then we have a *deterministic Turing machine* (D-TM or just
TM), the machine originally envisioned by Turing.

We now define still another type of machine. Let a <u>BA-TM</u> be a PA-TM
restricted so that there is only one universal state; and furthermore
the universal player can never write on the common tapes, nor move the
heads of the common tapes. Note that the computation game of a BA-TM
is a blindfold game. Thus we have defined for each *game type* g in
$\mathcal{G}$ = {incomplete information, blindfold, perfect information, nondeter-
ministic, deterministic} a corresponding *machine type* m(g) in
$\mathcal{M}$ = {private alternating, blind alternating, alternating, nondeterministic,
deterministic} with computation game of type g.

The winning strategies of computation games can be recursively
enumerated, and thus the language of each PA-TM and BA-TM is recursively
enumerable. Also, the D-TMs accept all the recursively enumerable sets
and each D-TM is a PA-TM and a BA-TM. Hence we have

THEOREM 2.1. The PA-TMs and BA-TMs each accept precisely the
recursively enumerable sets.

Next we consider the computational complexity of PA-TMs and BA-TMs.

M has *space bound* $S(n)$ (*time bound* $T(n)$, *alternation bound* $A(n)$,
respectively) *if for each input string* $w \in \Sigma^n$ accepted by M, there is an
accepting subtree $T'$ such that no tape has more than $S(n)$ nonblank
cells on any configuration (each computation sequence $\pi \in T'$ has

$\leqslant T(n)$ moves, each $\pi \in T'$ has $\leqslant A(n)$ alternations, respectively). Thus computation game $G_M$ has space bound $O(S(n+O(1)))$ (time bound $T(n+O(1))$, alternation bound $A(n+O(1))$, respectively) if $M$ has space bound $S(n)$ (time bound $T(n)$, alternation bound $A(n)$, respectively). By the usual tape encoding techniques (where we encode each $2c$ consecutive work tape cells as a 2c-tuple in a new tape alphabet), we have a constant space compression result:

THEOREM 3.2. For any constant $c > 1$ and a machine $M$ of machine type $g \in \mathscr{G}$ with space bound $S(n) \geqslant c$, there is a machine with space bound $S(n)/c$ that accepts the same language as $M$, and with the same machine type $g$ as $M$, and with no additional alternations.

We also have a constant speed-up result:

THEOREM 3.3. For any constant $c > 1$ and any machine $M$ of any type in $\mathscr{G}$, there is a machine of the same type as $M$ with time bound $T(n)/c$, that _ccepts the same language as $M$.

Proof. There is a constant $d$ upper bounding the number of next possible moves from any given position of $M$. Thus there are $\leqslant d^{20c}$ positions of $M$ reachable after $20c$ moves from any given position. We construct a simulating machine $M'$ of the same type as $M$. As in Theorem 3.2, we encode each $20c$ consecutive cells of $M$ as a 20c-tuple in the tape alphabet of $M'$. $M'$ will have a distinguished state associated with each of the $2^{d^{20c}}$ possible strategies of the existential and universal player within 20c moves from any given position. Given an input string $w \in \Sigma^n$, the simulation will proceed by having $M'$ first move one cell left, two cells right, and then one cell left on each of its tapes so as to determine the current relevant tape contents. Then the

existential player of M' is allowed, by a single state transition, to choose its strategy for the next 20c simulation steps of M (if no such strategy exists, M' rejects). If there were any alternations within these 20c simulation steps, then we allow the universal player of M' a single state transition to choose its moves within these 20c simulation steps of M (if no such moves are possible for the universal player, then M' accepts). If there has been no rejection or acceptance within these 20c simulated steps of M, then M' makes four additional moves of the tape heads: (left, twice right, and left again) to update the tapes, and then the simulation proceeds as above. M' makes 10 steps for every 20c steps of M, and the total time bound of M' is $\leqslant \lceil T(n)/c \rceil$. Furthermore, M' accepts w if M accepts w. □

Next we show that the computation games of various tapes of machines are universal for the corresponding classes of games. Fix some functions $S(n) \leqslant \log n$ and $A(n)$ and let g be a game type in $\mathscr{G}$. Let $\mathscr{C}$ be the class of games of fixed game type g with space bound $S(n)$ and alternation bound $A(n)$. Let us assume that the set of positions derived by a single move from any given position of length n, can be computed in deterministic space $MS(n)$. For each game $G = (POS, \vdash)$ of $\mathscr{C}$, let $B_G$ be the deterministic log space mapping from positions in POS to their binary string representation. Let $N_G$ be a binary string encoding the deterministic space $MS(n)$ next move transducer for $\vdash$.

Clearly, there is a machine $M_{\mathscr{C}}$ such that for each game $G \in \mathscr{C}$ and position p of G, M accepts $(N_G, B_G(p))$ iff player 1 has a winning strategy in G from initial position p. Thus $M_{\mathscr{C}}$ decides the outcome of all the games of $\mathscr{C}$. Furthermore, $M_{\mathscr{C}}$ has corresponding

machine type m(g) (i.e., its computation game is of type g) has tape
alphabet {0,1,b,#}, space bound S(n) + MS(S(n)), and alternation bound
A(n). If MS(S(n)) = O(S(n)) then by Theorem 3.2, $M_{\mathscr{C}}$ need to have
only space bound S(n).

Thus we have shown:

THEOREM 3.4. If MS(S(n)) = O(S(n)) then the computation game $G^{M_{\mathscr{C}}}$
is a universal game for the game class $\mathscr{C}$.

By applying the space compression Theorem 3.2, we have

COROLLARY 3.4. For each game type g ∈ $\mathscr{G}$, if $\mathscr{R}$ is the class of
reasonable games (i.e., with space bound S(n) = n) of type g, then there
is a linear space bounded machine $M_{\mathscr{R}}$ of corresponding type m(g) such
that $G^{M_{\mathscr{R}}}$ is a universal game for $\mathscr{R}$.

## 4. DECISION ALGORITHMS FOR SPACE BOUNDED GAMES

It is easy to show

THEOREM 4.1. Any deterministic (nondeterministic, respectively) game with space bound $S(n) \geqslant \log n$ can be decided in deterministic (nondeterministic, respectively) space $O(S(n))$.

By applying the result of [Savitch, 1970], we can easily show:

COROLLARY 4.1. Any nondeterministic game with space bound $S(n) \geqslant \log n$ can be decided in deterministic space $O(S(n)^2)$.

We consider now in turn decision algorithms for deciding games of perfect information, then games of incomplete information, and finally blindfold games.

### 4.1 Deciding a Game of Perfect Information

Theorem 4.2. For any $S(n) \geqslant \log n$, the outcome of any game $G$ of perfect information with space bound $S(n)$ can be decided in deterministic time $2^{O(S(n))}$.

This result will be utilized in Section 4.2. For completeness, we give here an algorithm similar to a procedure previously given by [Chandra, Kozen, and Stockmeyer, 1978] for determining acceptance of an alternating machine with a space bound. We assume $S(n)$ is constructible (else try the below with $S(n) = 0, 1, \ldots$).

Let $G = (POS, \vdash)$. Given an initial position $p_I$ of length $n$, we construct a set $POS(p_I)$ of all positions reachable by moves of $G$ from

from $p_I$ and with space $\leqslant S(n)$. Since $G$ has position size bound $S(n)$ there must be a constant $c$, independent of $n$, such that $|POS(p_I)| \leqslant c^{S(n)}$.

We will also construct a sequence of mapping $s$ from $POS(p_I)$ to $\{$ <u>true</u>,<u>false</u> $\}$. Initially, let $\ell(p) = $ <u>false</u> for each $p \in POS(p_I)$. We then compute a new mapping $f(\ell)$ such that for each $p \in POS(p_I)$,

$$
\begin{array}{llll}
f(\ell)(p) & = \underline{\text{false}} & \text{if} & p \in W \wedge POS_1 \\[2mm]
& = \displaystyle\bigvee_{p \vdash p'} \ell(p) & \text{if} & p \in POS_1 - W \\[2mm]
& = \underline{\text{true}} & \text{if} & p \in W \wedge POS_0 \\[2mm]
& = \displaystyle\bigwedge_{p \vdash p'} \ell(p') & \text{if} & p \in POS_0 - W
\end{array}
$$

Let $\ell^*$ be the mapping derived by repeating applying $f$ to $\ell$ until there is no change. This requires at most $|POS(p_I)|$ iterations and $2^{O(S(n))}$ deterministic time per iteration, since we have assumed that the next moves in all games are computable in linear space. Thus $2^{O(S(n))}$ total time is required. Then we can show there is a 1-1 correspondence between Markov strategies $\sigma$ of player 1 and labelings $\ell^*$ constructed by the above process. In particular, the positions mapped by $\ell^*$ to <u>true</u> correspond to the positions appearing in winning plays induced by some such $\sigma$, and vice versa. Thus we can show $\ell^*(p_I) = $ <u>true</u> iff player 1 has a winning Markov strategy for $p_I$. By Proposition 2.2, Markov strategies suffice. □

Since any labeling $\ell^*$ of Theorem 4.2 with $\ell^*(p_I) = \underline{true}$ corresponds to a winning Markov strategy whose plays are each of length $\leq 2^{O(S(n))}$, we have

COROLLARY 4.2. If $G$ is a game of perfect information with space bound $S(n) \geq \log n$ then $G$ has time bound $2^{O(S(n))}$.

## 4.2 Eliminating Incomplete Information from a Game

We now give a powerset construction for transforming a game $G = (POS, \vdash)$ of incomplete information into a game $G^+ = (POS^+, \vdash^+)$ of perfect information whose positions are *sets* of positions of $G$. (The construction is somewhat reminiscent of the subset construction in finite state automata.) Our decision algorithms will rely on this construction, which entails an exponential blow-up in space complexity. In Section 5.3, we show that, in the worst case, such a complexity blow-up must occur.

Fix some initial position $p_I \in POS$. We will assume that the set of positions reachable by moves from $p_I$ is finite. For each play prefix $\pi$ of $G$ we construct a position $P(\pi)$ of $G^+$ with common portion the set $\{last(\pi') \mid \pi' \text{ is a play prefix with } vis_1(\pi) = vis_1(\pi')\}$. (This is the set of current possible positions after $\pi$, from player 1's point of view, by viewing only $vis_1(\pi)$.) Let the private portions of $P(\pi)$ be *null* (thus $G^+$ is a game of perfect information) and let the next player to move in $P(\pi)$ be the same as in $last(\pi)$. Note that if $vis_1(\pi) = vis_1(\pi')$, then the next player to move in $last(\pi)$ is the same as in $last(\pi')$. Hence

$$P(\pi) = P(\pi') \quad \text{iff} \quad vis_1(\pi) = vis_1(\pi') .$$

We allow no next move from $P \in POS^+$ if it is player 1's turn to move and $P = P(\pi)$ for some play prefix $\pi$ with $last(\pi) \in W$. (Thus player 2 wins at $P$ if $P = P(\pi)$ for some $\pi$ which is winning for player 2.) Otherwise, we let $P \vdash^+ P'$ be a move of $G^+$ if there exists play prefixes $\pi, \pi'$ of $G$ such that $P = P(\pi)$, $P' = P(\pi')$ and $\pi'$ is a child of $\pi$. (Thus, moves of $G^+$ from $P$ simulate all possible moves of $G$ from any position $last(\pi)$ when $P = P(\pi)$.) Fix $P_I = P(p_I)$ to be the initial position of $G^+$.

THEOREM 4.3. Player 1 has a winning strategy in $G$ from initial position $p_I$ iff player 1 has a winning strategy in $G^+$ from $P_I$.

Proof. We establish a 1-1 correspondence between winning strategies of $G$ and winning Markov strategies of $G^+$.

CASE 1. Let $\sigma$ be a winning strategy for player 1 in $G$. For each play prefix $\pi$ of $G^+$, where it is player 1's turn to move at $last(\pi)$, let $\sigma^+(\pi) = \pi P(\sigma(\pi))$ for any play prefix $\pi$ of $G$ such that $p(\pi) = last(\pi)$. $\sigma^+$ is now shown by contradiction to be a winning Markov strategy for $G^+$. Suppose $\pi$ is a play of $G^+$ induced from $\sigma^+$ but $\pi$ is not winning for player 1. Then there is a play $\pi$ of $G$ induced from $\sigma$ whose $P(\pi) = last(\pi)$, and such that $\pi$ is not winning for player 1. But this contradicts our assumption that $\sigma$ is winning.

CASE 2. On the other hand, let $\sigma'$ be a winning strategy for player 1 in $G^+$. By Proposition 2.2, we can assume without loss of generality that $\sigma'$ is a Markov strategy. For each play prefix $\pi$ of $G$, where it is player 1's turn to move in $last(\pi)$, let $\sigma(\pi)$ be a child

of $\pi$ such that $\sigma'(\pi) = \pi P(\sigma(\pi))$ for any play prefix $\pi$ of $G^+$ such that $P(\pi) = last(\pi)$. Again, $\sigma$ can easily be shown by contradiction to be a winning strategy for $G$. $\qquad\qquad$ □

Note that if $G$ has space bound $S(n) \geqslant \log n$, then $G^+$ has space bound $2^{O(S(n))}$; thus we have an exponential blow-up in complexity by our method of eliminating incomplete information. By applying Corollary 4.2 to Theorem 4.3, we have

COROLLARY 4.3. If $G$ is a game of incomplete information with space bound $S(n) \geqslant \log n$, then $G$ has time bound $2^{2^{O(S(n))}}$.

Theorem 4.3 motivates a decision algorithm for a game $G = (POS, \vdash)$ of incomplete information, with initial position $p_I$. Our algorithm will utilize an alternating Turing machine.


Algorithm A:

$\qquad P \leftarrow \{p_I\}$

$\qquad$ WHILE true DO

$\qquad\qquad P' \leftarrow \{p' \mid p \vdash p', p \in P\}$

$\qquad\qquad W(P) \leftarrow \{p \in P \mid p \vdash p' \text{ for no } p'\}$

$\qquad\qquad V \leftarrow \{vis_1(p) \mid p \in P'\}$

$\qquad\qquad$ IF $P \subseteq POS_1$ THEN

$\qquad\qquad\qquad$ BEGIN

$\qquad\qquad\qquad\qquad$ COMMENT player 1's move

$\qquad\qquad\qquad\qquad$ IF $W(P) \neq \emptyset$ THEN L1: REJECT

$\qquad\qquad\qquad\qquad$ ELSE L2: $v \leftarrow$ an existentially chosen element of $V$

$\qquad\qquad\qquad$ END

    __ELSE__

        __BEGIN__

            __COMMENT__ player 0's move with $P \subseteq POS_0$

            __IF__ $W(P) = P$ __THEN__ L3: __ACCEPT__

            __ELSE__ L4: $v \leftarrow$ a universally chosen element of $V$

        __END__

        $P \leftarrow \{p' \in P' \mid vis_1(p') = v\}$

    __OD__

Intuitively, the algorithm tests for the existence of a winning
strategy for player 1 by simulating all possible plays by all possible
strategies simultaneously. Trial strategies are extended existentially,
one step at a time. At each step all possible moves of player 2 are
simulated to determine whether the strategy is adequate so far. If not,
it is rejected; if so, it is continued to be extended. The invariant of the
while loop is that $P$ is a set of the form

$$\{last(\pi') \mid vis_1(\pi) = vis_1(\pi'), \ \pi' \text{ is a play prefix from } p_I\}$$

for some play prefix $\pi$ from $p_I$ in $G$. Thus, $P$ is equivalent to the
common portion of a position of the game $G^+$. This loop invariant also
implies that either $P \subseteq POS_1$ or $P \subseteq POS_0$, since it can be determined
from the visible portion of a position whose turn it is.

We have four conditions within the body of the while statement. In
the case L1 is reached, it is player 1's turn and player 0 had a sequence
of moves against this partial strategy that lead to a position
$p \in POS_1 \wedge W$. In this case we must reject, since the partial strategy has
been shown inadequate. At L2, it is player 1's move and he has a next move

from every possible position. In this case the strategy is extended

existentially one move step in all possible ways. In the cases L3 or L4

are reached, player 0 has the initiative. At L3, he has no next move,

so this branch of the trial strategy is winning for player 1. At L4,

player 0's next move is chosen universally among all possible, reflecting

the fact that any strategy of player 1 must fail them all.

Thus Algorithm A implements the game $G^+$ by use of an alternating

machine. Algorithm A accepts exactly when there is a winning strategy for

player 1, since the algorithm establishes a one to one correspondence

between these winning strategies and finite accepting subtrees of the

computation tree of the alternating machine.

THEOREM 4.4. The outcome of any game $G$ of incomplete information

with space bound $S(n)$ can be decided by an alternating machine with

space bound $2^{O(S(n))}$.

Proof. Since there are no more than $2^{O(S(n))}$ positions of $G$

reachable from $p_I$, Algorithm A can easily show to be implemented by an

alternating machine with space bound $2^{O(S(n))}$. □

By Theorems 4.2 and 4.4 we have:

THEOREM 4.5. The outcome of any game $G$ of incomplete information

with space bound $S(n)$ can be decided in deterministic time $2^{2^{O(S(n))}}$.

## 4.3  A Decision Algorithm for Blindfold Games

We show here

THEOREM 4.6.  Any blindfold game  G  with space bound  S(n)  can be decided in nondeterministic space  $2^{O(S(n))}$.

Proof. 'Let  $G = (POS, \vdash)$  as in the proof of Theorem 4.4.· Let  $\mathcal{V}$  be the set of all  V  of the form  $\{vis_1(p) \mid p \in P\}$  such that  P  is a set of positions in POS reachable from  $p_I$.  Let  D  be a mapping such that  $\forall V \in \mathcal{V},$  $D(V) \in V.$

Let Algorithm A be modified to  A'  by substituting  $v \leftarrow D(V)$  at step L4. Also enclose the while loop by another while loop which iteratively assigns  D  all possible values.  For Algorithm A to accept, the inner loop must accept on all iterations of the new outer loop.  The resulting Algorithm A' is *nondeterministic* (since we utilize only existential choice). We claim that if  G  is blindfold, then Algorithm A' accepts iff player 1 has a winning strategy.  To see this, we simply observe that since the game is blindfold, the moves chosen by player 1 in its winning strategy ·are oblivious to any moves by player 0.                               ◻

## 4.4  Games with Both Alternation and Space Bounds

THEOREM 4.7.  For any game  G  of perfect information with alternation bound  A(n)  and space bound  $S(n) \geqslant \log n$, the outcome of  G  can be decided in deterministic spaces  $(A(n) + S(n))S(n)$.

Proof. By Theorem 3.2 we can show that the outcome of G can be decided by an alternation machine M with alternation bound $A(n)$ and space bound $S(n)$. Borodin has shown (see [Chandra, Kozen, and Stockmeyer, 1978]) that the acceptance problem for M can be decided in space $(A(n) + S(n))S(n)$.                                    □

Now let G be a game of incomplete information with alternation bound $A(n)$ and space bound $S(n) \geqslant \log n$. Fix an initial position of length n. By Theorem 4.3, the game $G^+$ of perfect information has the same outcome as G, and by Corollary 4.3, $G^+$ has space bound $2^{O(S(n))}$. But $G^+$ has the same alternation bound $A(n)$ as G. Thus by Theorem 4.7,

THEOREM 4.8. For any game G of incomplete information with alternation bound $A(n)$ and space bound $S(n) \geqslant \log n$, the outcome of G can be decided in deterministic space $(A(n) + 1)2^{O(S(n))}$.

# 5. LOWER BOUNDS ON THE COMPLEXITY OF SPACE BOUNDED GAMES

To derive our lower bounds, we use the technique of encoding computations of a standard type of machine into one of our new types of machines. Then we can apply known hierarchy results known for the standard type of machine, to obtain the desired lower bounds for our new types of machines.

## 5.1 Lower Bounds on Games of Perfect Information

This technique was utilized by [Chandra, Kozen, and Stockmeyer, 1980] to obtain lower bounds for games of perfect information. They show:

THEOREM 5.1. For each $S(n) \geqslant \log n$

$$ASPACE(S(n)) \supseteq DTIME(EXP(S(n)))$$

(see definition of EXP in the Introduction).

By applying their version of Theorem 4.1, they have

COROLLARY 5.1. For each $S(n) \geqslant \log n$

$$ASPACE(S(n))) = DTIME(EXP(S(n))) \ .$$

This is an elegant characterization of the power of space bounded alternation. We aim to derive such characterizations for private and blind alternations.

## 5.2 Lower Bounds for Blindfold Games

THEOREM 5.2.  For each $S(n) \geqslant \log n$,

$$BASPACE(S(n)) \supseteq NSPACE(EXP(S(n)))).$$

**Proof.** Let $M$ be a N-TM with an input string $\omega \in \Sigma^n$. We assume $M$ has a constructible space bound $c^{S(n)}$ for some constant $c > 0$ (if it is not constructible, we try $S(n) = 0,1,\ldots$). Let $\vdash$ be the next move relation of $M$. It will be useful to assume that for each position $p$ of $M$, that is neither accepting nor rejecting, there are exactly $d$ next moves (where $d$ is a constant dependent only on $M$) $\vdash_1(p),\ldots,\vdash_d(p)$. We consider the configurations of $M$ to be strings over a finite alphabet $\Delta$. Let $D = \{1,\ldots,d\}$ be considered symbols disjoint from $\Delta$ and let $\Delta' = \Delta \cup D$.

We now construct a BA-TM $M_1$ with space bound $S(n)$. The players will alternate on each move. $M_1$ will require a unique state for each symbol in $\Delta'$. Let the existential player of $M_1$ choose (by entering the appropriate states) a string of the form $p_0 r_1 p_1 r_2 \cdots r_k p_k$ where $r_1,\ldots,r_k \in D$ and $p_0,\ldots,p_k \in \Delta^*$. Let the universal player of $M_1$ choose to privately (by use of a private tape) verify one of the following conditions is violated.

   (i)  $p_0$ is the initial configuration of $M$

   (ii)  $p_k$ contains the accepting state of $M$

   (iii)  $p_i = \vdash_{r_i}(p_{i-1})$ for $i = 1,\ldots,k$.

Note that if (i), (ii), and (iii) all hold then the string chosen by the existential player of $M_1$ is an accepting computation (if the $r_i$ are

ignored). This is the goal of the existential player of $M_1$. The universal player of $M_1$ is trying to verify that the string chosen by the existential player is *not* an accepting computation.

(Note that it is essential that the universal player of $M_1$ *privately* choose to verify (i), (ii), or (iii), or otherwise the existential player of $M_1$ could "cheat" by observing which of (i), (ii), or (iii) are tested and then varying the choice of string $p_0 r_1 p_1 r_2 \cdots r_k p_k$ so that not all of (i), (ii) and (iii) hold for any choice of the string.)

To verify (i) is violated, the universal player of $M_1$ may utilize log n cells of a private tape for a pointer to symbols of the input string w. It is trivial to verify the case (ii) is violated. For the case (iii) it is useful to define for each $r$, $1 \leqslant r \leqslant d$, a function $F_r$: $\Delta' \times \Delta' \times \Delta' \times \Delta' \to \Delta'$, such that for each $a_{-1} a_0 a_1 a_2 \in \Delta'$, if $a_0 \in D$ then $F_r(a_{-1}, a_0, a_1, a_2) = a_0$ and otherwise if $a_{-1} a_0 a_1 a_2$ are the $j-1, j$, $j+1, j+2$ symbols of string $rpr'$, then $F_r(a_{-1} a_0 a_1 a_2)$ is the j-th symbol of the string $r'p'r''$ where $p' = \vdash_{r'}(p)$ for configurations $p$, $p'$ and $r, r', r'' \in D$.

To verify (iii) is violated, let the universal player choose an integer $i$, $1 \leqslant i \leqslant k$ and some $j$, $1 \leqslant j \leqslant \text{length}(p_i)$ and store on a private tape $a_{-1} a_0 a_1 a_2$ which are the $j-1, j, j+1, j+2$ symbols of $r_{i-1} p_{i-1} r_i p_i$. The universal player must then test that $F_{r_i}(a_{-1} a_0 a_1 a_2)$ is the j-th symbol of the string $r_i p_i r_{i+1}$. The total space cost is thus $S(n)$, since $j$ and $k$ are both $\leqslant 2^{O(S(n))}$. We let $M_1$ accept just if the universal player cannot verify either (i), (ii), or (iii) has been violated. Thus $M_1$ accepts iff there exists an accepting computation $p_0 p_1 \cdots p_k$ of M. Clearly $M_1$ is blindfold since the moves of the existential players are completely oblivious to the moves of the universal players.     □

By combining Theorems 4.6 and 5.2 we have

COROLLARY 5.2.  For each  $S(n) \geqslant \log n$,

$$BASPACE(S(n)) = NSPACE(EXP(S(n))).$$

## 5.3  Lower Bounds for Games of Incomplete Information

The reader may inquire:  did the proof of Theorem 5.2 utilize the full power of private alternating machines?  Indeed, it did not, since the simulation game was blindfold.  The following Theorem uses a similar construction, but also employs the dynamic interaction between the existential and universal player possible in general games of incomplete information.

THEOREM 5.3.  *For each*  $S(n) \geqslant \log n$,

$$PASPACE(S(n)) \supseteq ASPACE(EXP(S(n))).$$

Proof.  Let  M  be an A-TM with input string  $\omega \in \Sigma^n$.  We assume  M has constructible space bound  $c^{S(n)}$,  if for some constant  $c \geqslant 1$  (otherwise we try  $S(n) = 0, 1, \ldots$).  Let  $\Delta, \Delta'$,  and  F  be defined just as in Theorem 5.2.  The proof is similar, however, here we construct in deterministic  $\log n$  space a PA-TM  $M_2$  with space bound  $S(n)$  which accepts iff  M  accepts.

We will require again a unique state of  $M_2$  for each symbol of  $\Delta'$;  all other states will be associated with a null symbol.  We also again let the players alternate on each move.  The players will choose (by entering the appropriate states) a string of the form  $p_0 r_1 p_1 r_2 \cdots r_k p_k$

where $r_1, \ldots, r_k \in D$ and $p_0, \ldots, p_k \in \Delta^*$. All these symbols will be chosen by the existential player, except that if $p_{i-1}$ contains a universal state, then the universal player publically chooses $r_i \in D$ by writing $r_i$ on a public tape (this has the effect of creating $d$ branches on the game tree; since the subsequent choice of $p_i$ by the existential player may be very dependent on observation of the universal player's choice of $r_i$). Again we require the universal player to privately (by use of a private tape) attempt to verify that one of the cases (i), (ii), or (iii) is violated.

Note that if the cases (i), (ii), (iii) hold for each choice of the $r_i$'s then the existential player of $M_2$ has chosen a set of strings which (if the $r_i$ symbols are ignored) are an accepting subtree (i.e., these strings are the accepting computation sequences induced by a winning strategy for the existential player in the game $G^M$). This is the goal of the existential player of $M_2$, and we let $M_2$ accept if this goal is achieved. Otherwise, if the universal player finds a violation of (i), (ii), (iii), (iv) then $M_2$ rejects. $\quad\square$

Combining Theorems 4.5, 5.1 and 5.3 we have:

COROLLARY 5.3. For each $S(n) \geqslant \log n$,

$$PASPACE(S(n)) = ASPACE(EXP(S(n)))$$
$$= DTIME(EXP(EXP(S(n)))).$$

As a consequence of Corollary 3.4, and the results of this section, we have

(1)   a space $n$ bounded PA-TM $M$ whose computation game $G^M$ is universal for all reasonable games.

(2) A space $n$ bounded BA-TM M' whose computation game $G^{M'}$ is

universal for all reasonable blindfold games.

By the hierarchy theorem for deterministic time complexity [Hartmanis and Stearns, 1965] we have

COROLLARY 5.4. There is a $c > 1$ such that if any D-TM decides the outcome of $G^M$ in time $T(n)$, then $T(n) > 2^{c^{n/\log n}}$.

By space hierarchy results,

COROLLARY 5.5. There is a $c > 1$ such that if any D-TM decides the outcome of $G^{M'}$ in space $S(n)$, then $S(n) > c^{n/\log n}$.

## 6. TIME BOUNDED BLIND AND PRIVATE ALTERNATING MACHINES

Let $\Sigma_{A(n)}^{T(n)}$ be the class of languages accepted by alternating machines with time bound $T(n)$, with alternation bound $A(n)$, and existential initial state.

[Chandra, Kozen, Stockmeyer, 1979] use this notation and give a result they attribute to Borodin:

THEOREM 6.1. For each $T(n) \geqslant n$,

$$DSPACE(T(n)) \subseteq \Sigma_{A(n)}^{T(n)} \subseteq NTIME(A(n) \ T(n)) \ .$$

We now characterize the time complexity of blind and private alternating machines in terms of the time complexity of alternating machines.

THEOREM 6.2. For each $T(n) \geqslant n$,

$$BATIME(T(n)) = \Sigma_2^{T(n)} \ .$$

Proof. Let $M$ be a BA-TM with time bound $T(n) \geqslant n$ and input string $\omega \in \Sigma^n$. Since the existential player of $M$ is oblivious to any move by the universal player of $M$, it might just as well have chosen its moves at the start of the computation, and stored them into a consecutive sequence of tape calls. This can be done in time $T(n)/2$ by Theorem 3.3, if we augment the tape alphabet so that each pair of moves of the existential player is represented by a distinct symbol. Next, we let the universal player choose all its moves and attempt to verify the resulting play is not accepting. This can also be done by Theorem 3.3

in time $T(n)/2$ using the augmented tape alphabet. Thus the resulting machine $M'$ has time bound $T(n)$ and accepts just the strings accepted by $M$. Note that the moves of the existential player of $M'$ proceed all the moves of the universal player. Thus, all portions of position of the universal player can be considered common, so $M'$ is an alternating machine. Thus we have shown $\text{BATIME}(T(n)) \subseteq \Sigma_2^{T(n)}$. (Note that this simulation is not particularly space efficient since $M'$ may now require at least space $T(n)/2$.)

To show $\Sigma_2^{T(n)} \subseteq \text{BATIME}(T(n))$, we first observe that if $M_1$ is an A-TM where all the moves of the existential player proceed all moves of the universal player then the existential player is oblivious to any subsequent moves of the universal player. If $M_1$ has time bound $T(n)$, then since it has only one alternation $M_1$ can be speeded up by a factor of two to $T(n)/2$ without introducing any further alternations. Let $M_2$ be the Ba-TM derived from $M_1$ by introducing a new private tape for each original tape on which the universal player did any writing or head movement operations. Each tape operation of the existential player must be simulated, in the next succeeding step, by the universal player on these new private tapes. This slows down the simulation time by a factor of two, to time $T(n)$, and introduces $T(n)/2$ alternations. The resulting blind alternating machine $M_2$ accepts just the strings accepted by $M_1$. □

THEOREM 6.3. For any $T(n) \geqslant n$,

$$\text{PATIME}(T(n)) = \text{ATIME}(T(n)) \quad .$$

Proof. (This result was first given in [Peterson and Reif, 1979] in a more general context of multi-player games.) First observe that any A-TM is a PA-TM, so $\text{PATIME}(T(n))$ contains $\text{ATIME}(T(n))$. On the other

hand, let M be a PA-TM with time bound $T(n)$ and input string $\omega \in \Sigma^n$. We can assume a constant $d$ bounding the maximum number of common portions of positions possible from a single position of M. We construct an A-TM M' which simulates M in two stages.

We require a set $\Gamma'$ of $d+1$ special new tape symbols for M', one for each set of next moves of M which are indistinguishable to the existential player, and also one distinguished symbol designating a "pass" move. In the first stage of the simulation, the existential and universal players alternatively write symbols of $\Gamma'$ on consequential calls of a new tape of M'. The existential player is allowed to terminate this stage at any time. In the next stage, the universal player attempts to verify that there is some play $\pi$ of M from the initial position and consistent with previously chosen moves, such that $\pi$ is not winning for the existential player. If so, the machine M' rejects, and otherwise M' accepts. The total time for these two phases is $3T(n)$, but this can be speeded up to $T(n)$ by Theorem 3.3. □

## 7.  UNIVERSAL GAMES ON PROPOSITIONAL FORMULAS

In this section we construct various propositional formula games
which are universal for reasonable games.  These games and the reductions
between them are generalizations of work on games of perfect information
in [Stockmeyer and Chandra, 1979].

Boolean variables take on values 1, 0 representing _true_, _false_, respectively.
Let a _literal_ be a boolean variable or its negation.  Let a propositonal formula
F  be in k-_conjunctive (disjunctive) normal form_ if  F  consists of a
conjunction (disjunction) of formulas  $F_1, F_2, \ldots, F_j$  with each  $F_i$  a
disjunction (conjunction, respectively) of at most  k  literals.

We now list 3 games on propositional formulas which are universal for
all reasonable games.  The game  $G^3$  is the game PRIVATE-PEEK described in
the Introduction.

(1)  Let  $\underline{G}^1$  be the game in which a position contains a propositional
formula  $F(X, Y^C, Y^{P0}, Y^{P1}, a, s)$  in 5-conjunctive normal form, with  $X^C, Y^{P0}, Y^{P1}$
each sequences of variables and  a,s  individual variables,  and also a truth
assignement to its variables.  The formula  F  and the truth assignment
to the variables of  $X, Y^C, a, s$  are common to both players 1 and 0, but the
truth assignment to the variables of  $Y^{P0}, Y^{P1}$  are private to player 0.

Player 1 moves by setting  a  to 1 and choosing a new truth assignment
for the variables of  X.  Player 0 moves by  (a) setting  a  to 0,
(b)  setting  s  to the complement of its previous truth assignement,
(c)  choosing a new truth assignment for the variables of  $Y^C, Y^{Ps}$.  The
formula  F  is not modified by these moves, except for the changes in the
truth assignment to its variables.  The loser is the first player whose
move yields a truth assignment for which the formula  F  is _false_.

(2) Let $\underline{G}^2$ be the game in which each position contains formulas $WIN_1(U,V^C,V^P)$ and $WIN_0(U,V^C,V^P)$ and truth assignements to the sequences of variables of $U,V^C,V^P$.

The formulas $WIN_1$ and $WIN_0$ and truth assignments to variables $U \cup V^C$ are viewed commonly by both players, but the truth assignment to the variables of $V^P$ are private to player 0. Player 1 moves by changing the truth assignment to at most one variable of $U$, while player 0 moves by changing at most one variable of $V^C,V^P$. Player $a \in \{0,1\}$ *wins* if formula $WIN_a$ is <u>true</u> after a move by player $a$.

(3) Let $\underline{G}^3$ be the game in which a position consists of a propositional formula $F'(U,V^C,V^P)$ in disjunctive normal form and a truth assignment to the variables of the sequences $U,V^C$ and $V^P$. The formula $F'$ and truth assignments to the variables of $U,V^C$ are viewed commonly by both players, but the truth assignment to the variables $V^P$ is private to player 0. Players move as in game $G^2$. A player wins if after his move the formula is <u>true</u>.

NOTE: $G^3$ is identical to the game PRIVATE-PEEK given in the Introduction of this paper, since each variable be identified with the "in" position of a unique plate of PRIVATE-PEEK, with the complement of that variable associated with the "out" position of the plate, and each clause corresponding to a set of positions of plates where the holes are vertically aligned so it is possible to peek from the top through to the bottom.

THEOREM 7.1. $G^1$, $G^2$, $G^3$ are all universal for reasonable games.

<u>Proof.</u> To show $G^1$ is universal for reasonable games, we consider the computation game $G^M$ known by Corollary 3.4 to be universal for all reasonable games where $M$ is space $n$ bounded PA-TM. Let $\omega \in \Sigma^n$ be an input string to $M$. We encode each position of $G^M$ as a bit vector of

length $n' = k_1 \cdot n$ (where $k_1$ depends only on the size of the tape alphabet of M), so that bits $1, 2, \ldots, n_c$ are those of $vis_1(p)$ (the portions of p common to both the existential and universal players), and the bits $n_c + 1, \ldots, n'$ contain those portion of p private to the universal player.

Using the techniques of [Stockmeyer, 1975], we may construct a propositional formula $NEXT(Z_1, Z_2, T)$ where $Z_1, Z_2, T$ are sequences of variables of length $n'$, $n'$, $k_2$ (where $k_2$ is a fixed constant) and such that: if $Z_1$ encodes a position $p_1$ then there exists an assignment to the variables of T such that $NEXT(Z_1, Z_2, T)$ is <u>true</u> if and only if $Z_2$ encodes some position $p_2$ derived from $p_1$ by a move of M. The size of NEXT is linear in the input length n.

We introduce new sequences of variables $X, Y^C, Y^{P0}, Y^{P1}$ of length m, m, r, r where $m = n_c + k_2$ and $r = n' - n_c$. Let $Y = Y^C Y^{P0} Y^{P1}$. Let $X[a, b]$ denote $X(a), X(a+1), \ldots, X(b)$ for any $1 \leqslant a \leqslant b \leqslant m$.

For distinct $s, \bar{s} \in \{0, 1\}$, let $NEXT_{0, s}(X, Y)$ be the formula derived from $NEXT(Z_1, Z_2, T)$ by substituting $X[1, n_c]$, $Y^{P\bar{s}}[1, p]$ for $Z_1$, substituting $Y^C[1, n_c]$, $Y^{Ps}[1, r]$ for $Z_2$, and substituting $Y^C[n_c+1, n_c+k_2]$ for T. Also, let $NEXT_{1, s}(X, Y)$ be derived from $NEXT(Z_1, Z_2, T)$ by substituting $Y^C[1, n_c]$, $Y^{P\bar{s}}[1, r]$ for $Z_1$, substituting $X[1, n_c]$, $Y^{Ps}[1, r]$ for $Z_2$, and substituting $X[n_c+1, n_c+k_2]$ for T. If we consider player 1 to be identified with the existential player of M and player 0 to be identified with the universal player of M, then for each $a \in \{0, 1\}$, $NEXT_{a, s}$ defines legal moves by player a on switch variable $s \in \{0, 1\}$.

Now we consider the formula

$$F(X,Y^C,Y^{P0},Y^{P1},a,s) = (a \wedge s \rightarrow NEXT_{1,1}(X,Y)) \wedge (a \wedge \neg s \rightarrow NEXT_{1,0}(X,Y))$$

$$\wedge (\neg a \wedge s \rightarrow NEXT_{0,1}(X,Y)) \wedge (\neg a \wedge \neg s \rightarrow NEXT_{0,0}(X,Y))$$

$F$ can easily be put in 5-conjunctive normal form.

Given the initial configuration $P_I(\omega)$ of $M$ on input $\omega$, let $s = 0$ and $a$ be assigned the player whose turn it is in $P_I(\omega)$. Also, let the variables $Y^C[1,n_c]$ $Y^{P0}[1,n'-n_c]$ be assigned to encode $P_I(\omega)$ and let all other variables be assigned arbitrarily. Let formula $F$ and this truth assignment be the initial position $p_1$ of game $G^1$. Then player 1 wins game $G^1_{p_1}$ if and only if player 1 (the existential player) wins the computation game of $M$ if and only if $M$ accepts input $\omega$. Thus we have a log-space reduction from the acceptance problem for $M$ to the outcome problem for $G^1$, and we conclude by Corollary 3.4 that $G^1$ is universal for reasonable games. Next, we show the formula $G^2$ is also universal for reasonable games.

We now introduce sequences of variables $U^A$, $U^B$, $V^A$, $V^B$ of length $m' = 4m + 2r + 4$. Let $U = U^A \cdot U^B$ and let $V = V^A \cdot V^B$. The sequences of variables $X, Y$ defined in the previous construction will, in legal plays of our game $G^2$, be contained in $U, V$ as in Figures $4_{1,0}$, $4_{0,0}$, $4_{1,1}$, $4_{0,1}$. The private portion $V^P$ of $V$ is where $Y^{P0}$, $Y^{P1}$ are located, and $V^C$ contains other elements of $V$.

For each $s \in \{0,1\}$ and player $a \in \{0,1\}$, let $NEXT'_{a,s}(U,V)$ be the formula derived from formula $NEXT_{a,s}(X,Y)$ by substituting formulas as in Figure $4_{a,s}$. Again, player 1 (player 0) is identified with the existential (universal) states of $M$, and $NEXT'_{a,s}$ is used to define next-moves by machine $M$.
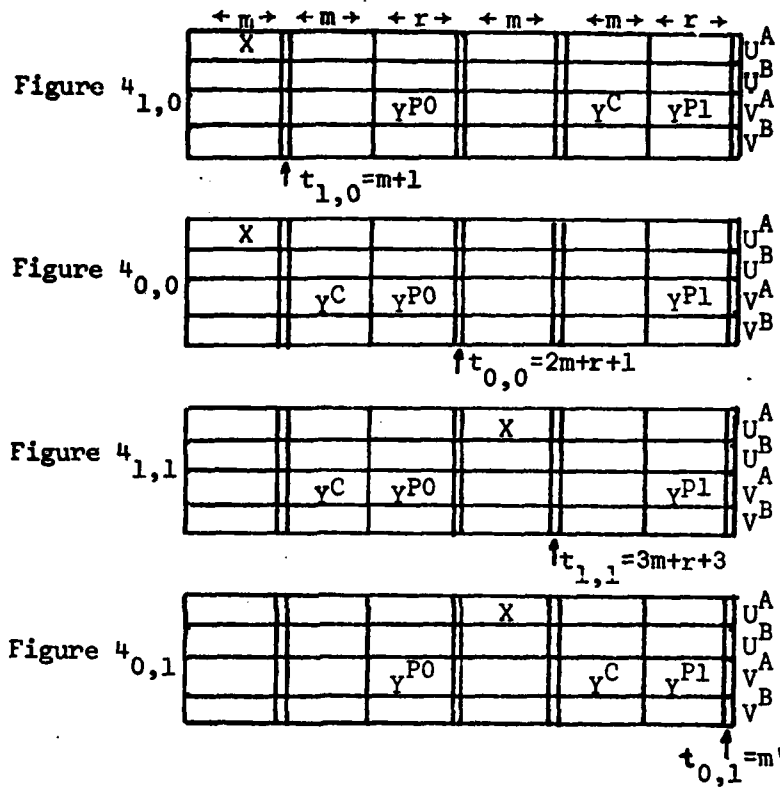
Figure 4

We describe a *legal play* such that if players 1 and 0 play legally, then player 1 wins if and only if $M$ accepts input $\omega$. Let a *legal cycle* be a play which satisfies the following *restriction* $\underline{L}$ for $i = 1, 2, \ldots, m'$:

player 0 changes the truth assignment of either $V^A(i)$ or $V^B(i)$.

player 1 changes the truth assignment of either $U^A(i)$ or $U^B(i)$.

Consider some $s \in \{0,1\}$ and distinct players $a, b \in \{0,1\}$. Within the legal cycle we also require *restriction* $\underline{L}'$ to hold: for each $i$, $(t_{a,s} \bmod m') < i \leqslant t_{b,s}$ player $a$ assigns variables so that $\text{NEXT}_{a,s}$ is $\underline{\text{true}}$ when $i = t_{a,s}$. Thus $M$ accepts input $\omega$ if and only if player 1 has a winning strategy within legal players satisfying restrictions $L$ and $L'$. The following construction forces legal play by both players.

We now introduce some notation for operations on sequences of $Z, Z'$ of boolean variables of length $m'$. Let $\oplus$ be exclusive-or and let

$$Z \oplus' Z' = (Z(1) \oplus Z'(1), \ldots, Z(n) \oplus Z'(n)),$$

and let

$$\Delta Z = (\neg(Z(n) \oplus Z(1)), Z(1) \oplus Z(2), \ldots, Z(n-1) \oplus Z(n)).$$

Also let $\text{TH-TWO}(Z) = \bigvee_{1 \leqslant i < j \leqslant m'} (Z(i) \wedge Z(j))$ be the threshold-two function. To simplify our formulas we define $U' = \Delta(U^A \oplus' U^B)$, and $V' = \Delta(V^A \oplus' V^B)$, which are sequences which locate boundaries between contiguous 0s or 1s. To detect illegal play we define:

$$\text{ILL}_1 = \text{TH-TWO}(U') \vee \bigvee_{1 \leqslant i \leqslant m'} (U'(i) \wedge V'(i) \wedge \neg V'(i-1))$$

$$\text{ILL}_0 = \text{TH-TWO}(V') \vee \bigvee_{1 \leqslant i \leqslant m'} (V'(i) \wedge U'(i+2) \wedge \neg U'(i))$$

Thus $ILL_b = \underline{true}$ just if player $b \in \{0,1\}$ has violated restriction $L$ for a legal cycle.

For each distinct players $a,b \in \{0,1\}$, let

$$ILL_b' = \bigvee_{s \in \{0,1\}} (U'(t_{a,s}) \wedge V'(t_{b,s}) \wedge \neg NEXT_{a,s}'(U,V)).$$

$ILL_b' = \underline{true}$ just if restriction $L'$ has been violated by player b. Finally let $WIN_a = ILL_b \vee ILL_b'$. Note that formula $WIN_0$ and $WIN_1$ can be put in disjunctive normal form.

Given input $\omega \in \Sigma^n$, let $p_1$ be the initial position of formula game $G^1$ defined previously.

Let the initial position $p_2$ of formula game $G^2$ contain formulas $WIN_0$, $WIN_1$ as defined above plus the initial truth assignment of $p_1$ as in Figure $4_{1,0}$. It can be shown player 1 wins game $G^2$ from initial position $p_2$ if and only if $M$ accepts $\omega$. Thus by Corollary 3.4, $G^2$ is also a formula game universal for all reasonable games.

Next, we give a log-space reduction from the outcome problem for formula game $G^2$ to the outcome problem for formula game $G^3$.

Let $u_1,\ldots,u_5,v_1,\ldots,v_5$ be variables not in $U$ or $V$ and set $\hat{V}^C = U \cdot (u_1,\ldots,u_5)$ and $\hat{V}^P = V^P \cdot (v_1,\ldots,v_5)$. Let

$$F'(U,\hat{V}^C,\hat{V}^P) = (WIN_0 \wedge (u_0 \vee v_1)) \vee (WIN_1 \wedge v_0) \vee (u_1 \wedge u_2 \wedge \neg v_1)$$

$$= ((v_1 \vee WIN_0) \wedge (u_2 \vee v_3)) \vee (u_4 \wedge u_5 \wedge \neg u_3)$$

$$\vee ((u_1 \vee WIN_1) \wedge (v_2 \vee u_3)) \vee (v_4 \wedge v_5 \wedge \neg u_3).$$

We let the initial position $p_2$ of formula game $G^3$ be given by the initial variable assignments of $p_2$ in $G^2$, and with the additional

variables $u_1,\ldots,u_5$; $v_1,\ldots,v_5$ set to 0. It is not difficult to show (see [Stockmeyer and Chandra, 1979]) that the game $G^3$ is winning for player 1 from initial position $p_3$ if and only if $G^2$ is winning for player 2 from initial position $p_2$. Thus, $G^3$ is another formula game universal for all reasonable games. □

Let $G^{2B}$, $G^{3B}$ be blindfold games derived from formula games $G^2$, $G^3$ by requiring that the common variable sequence $v^C$ of player 0 be empty. We claim that

THEOREM 7.2. $G^{2B}$, $G^{3B}$ are universal for all resonable blindfold games.

Proof. To show this, we need only note that if $M$ is restricted to BA-TM, then the universal player can never modify the common tape. Hence the common variables $v^C$ in our previous construction contain no information relevant to a configuration of $M$ (though they are useful to insure legal play) and hence the variables $v^C$ may be added to the variables $v^P$ private to player 0. The result then follows from our proof of Theorem 7.1.□

We have seen that the formula games $G^3$ and $G^{3B}$ are identical to the games PRIVATE-PEEK and BLIND-PEEK described in the introductory section. Thus we conclude by Theorems 7.1 and 7.2,

(1) PRIVATE-PEEK is a universal reasonable game.

(2) BLIND-PEEK is a universal reasonable blindfold game.

Note that our $\log n$ space reduction from the computation game $G^M$ to the game $G^3 = $ PRIVATE-PEEK has length bound $O(n^3)$. Thus by Corollary 5.4 and Theorem 7.1,

COROLLARY 7.1. There is a $c > 1$ such that if a D-TM decides the outcome of PRIVATE-PEEK in time $T(n)$, then $T(n) > 2^{c^{n^{1/3}}/\log n}$.

There is also a length bound $O(n^3)$ for our $\log n$ space reduction from a universal blindfold computation game to $G^{3B} = $ BLIND-PEEK. By Corollary 5.5 and Theorem 7.2,

COROLLARY 7.2. There is a $c > 1$ such that if a D-TM decides the outcome of BLIND-PEEK in space $S(n)$, then $S(n) > c^{n^{1/3}}/\log n$.

## 8. CONCLUSION

This paper has considered the computational complexity of *two* player games of incomplete information. Our general conclusion is that if the space is bounded by $S(n)$, then their outcome is an exponential more difficult to decide then for games of perfect information with space bound $S(n)$. Because of our lower bounds, our decision algorithms for games of incomplete information are asymptotically optimal.

It would be interesting to extend our results for the game of PRIVATE-PEEK to prove other games, such as "blindfold chess" are universal for all reasonable games of incomplete information. (The complexity of blindfold pursuit games on digraphs were considered in a previous draft of this paper [Reif, 1979].)

In [Peterson and Reif, 1979] we investigate the complexity of *multiple* player games of incomplete information. Our general conclusions for multiperson games with a position size bound $S(n)$ are:

(1) If the division of private information is not restricted, then the outcome problem is undecidable even for 3 player games;

(2) However, the multiplayer games are decidable if the private information is hierarchically divided among the players; and each additional player increases the complexity of the outcome problem by a further exponential.

[Petersen, 1980] applies these results to succinctness of string representation.

It is interesting to note that our technique of introducing private storage to an alternating machine, resulting in a PA-TM, could also be applied to another basic parallel machine type, such as a parallel PAM.

In that case each processor might have a private randomly accessible set of registers. Applications of multiplayer games of incomplete information to distributed multiprocessing problems are discussed in [Peterson and Reif, 1979] and a related multiprocess logic is described in [Reif and Peterson, 1980].

## Acknowledgments

## BIBLIOGRAPHY

Chandra, A.K., D.C. Kozen and L.J. Stockmeyer, "Alternation," Research
   Report RC 7489, IBM, Yorktown Heights, N.Y., Jan., 1978.

Even, S. and Tarjan, R.E., "A combinatorial problem which is complete in
   polynomial space," 7th Annual ACM Symp. on Theory of Computing, May,
   1976, Hershey, pp. 41-49.

Fraenkel, A.S., Garey, M.R., Johnson, D.S., Schaefer, T.J. and Yesha, Y.,
   "The complexity of checkers on an $N \times N$ board-preliminary report,"
   Proc. 19th IEEE Symp. on Foundations of Computer Science, Oct.
   1978, pp. 55-64.

Hartmanis, J. and R.E. Stearns, "On the computational complexity of
   algorithms," *Trans. Amer. Math. Soc.* 117 (1965), pp. 285-306.

Jones, N.D., "Blindfold games are harder than games with perfect inform-
   ation," *Bulletin of the European Assoc. for Theoretical Comp. Sci.*,
   No. 6, Oct. 1978, pp. 4-7.

Jones, N.D., "Space-bounded reducibility among combinatorial problems,"
   *J. Comput. System Sci.* 11, (1975), pp. 68-85.

Lichtenstein, D. and Sipser, M., "GO is Pspace Hard," *JACM* 27 (1980),
   pp. 393-401.

Lichtenstein, D. and Sipser, M., "Computing a perfect strategy for $n \times n$
   chess requires time exponential in N," 8th ICALP, Acre, Israel,
   (July 1981), pp. 278-293.

Peterson, G.L. and J.H. Reif, "Multiple Person Alternation," Proc. 20th
   IEEE Symp. on Foundations of Computer Science, October 1979.

Peterson, G.L., "Succinct representation random strings, and complexity
   classes," 21st Foundations on Foundations of Computer Science,
   (Oct. 1980), pp. 86-95.

Reif, J.H., "Universal two person games of incomplete information,"
   11th Annual ACM Symposium for Theory of Computing, 1979.

Reif, J.H. and G.L. Petersen, "A dynamic logic of multiprocessing with
   incomplete information," Proceedings 7th Annual ACM Symposium on
   Principles of Programming Languages, Las Vegas, (Jan. 1980),
   pp. 193-202.

Savitch, W.J., "Relationships between nondeterministic and deterministic tape complexities," *J. Comput. Sci.* 4, (1970), pp. 177-192.

Schaefer, T.J., "Complexity of some two person perfect-information games," *J. Comp. and Sys. Science* 16, No. 2, pp. 188-225.

Stockmeyer, L.J., "The polynomial-time hierarchy," *Theoretical Computer Science* 3 (1977), pp. 1-22.

Stockmeyer, L.J. and Meyer, A.R., "Word problems requiring exponential time: Preliminary report," Proceedings 5th ACM Symp. on Theory of Computing, 1973, pp. 1-9.

Stockmeyer, L.J. and Chandra, A.K., "Provably difficult combinatorial games," *SIAM J. Computing* 8, No. 2 (May 1979), pp. 151-174.